# "Private Prediction Strikes Back!" Private Kernelized Nearest Neighbors with Individual Rényi Filter (Supplementary material)

**Yuqing Zhu**[1] **Xuandong Zhao**[1] **Chuan Guo**[2] **Yu-Xiang Wang**[1]

[1]UC Santa Barbara
[2]FAIR / Meta AI

## 1 OMITTED PROOFS AND ALGORITHM IN SECTION 3

**Theorem 1.1** (Restatement of Theorem 3.1)**.** *Algorithm 3 satisfies* $(\alpha, B\alpha)$*-RDP for all* $\alpha \geq 1$.

*Proof.* The privacy analysis relies on individual RDP (Definition 2.4), which quantifies the maximum impact of adding or deleting a specific individual from any potential dataset to the prediction outcome, measured in terms of Rényi divergence.

We first demonstrate that only the selected neighbors have to account for their individual privacy loss. The decision rule for "being selected" is based on a comparison between the kernel weight and a data-independent threshold $\tau$, which is not influenced by any other private data points. Therefore, "unselected" neighbors do not incur any individual privacy loss.

For each selected neighbor $(x_i, y_i)$ at time $t$, its individual privacy analysis is broken down into two parts: the first part is the release of the number of neighbors $|\mathcal{N}_t|$, and the second part is the release of its label associated with the kernel weight.

Note that adding or removing one selected neighbor would only change $|\mathcal{N}_t|$ by 1, thus the individual RDP of releasing $|\mathcal{N}_t|$ at order $\alpha$ satisfies $\frac{\alpha}{2\sigma_1^2}$-RDP for all selected data. We next analyze the individual RDP of releasing the label. Fix a selected neighbor $z = (x_i, y_i)$, for all possible set of selected neighbors $\mathcal{N}_t = (z_1, ..., z_m)$ that include $z$, it holds that

$$D_\alpha^{\leftrightarrow}\Bigg( \big( \sum_{j\in\mathcal{N}_t} \kappa(x_j, q_t)\cdot y_j \big) + \mathcal{N}(0, \sigma_2^2 K_t \mathbb{I}_c) \Big|\Big| \big( \sum_{j\in\mathcal{N}_t\setminus z} \kappa(x_j, q_t)\cdot y_j \big) + \mathcal{N}(0, \sigma_2^2 K_t \mathbb{I}_c) \Bigg) \leq \frac{\kappa(x_i, q_t)^2 \alpha}{2\sigma_2^2 K_t}$$

by the definition of individual RDP.

Finally, The "delete" step in the algorithm ensures that the privacy loss for each private data point $(x_i, y_i)$ is bounded by a fixed value $B$, i.e., $\sum_{j=1}^{t} \left( (g_i + \frac{1}{2\sigma_1^2 \cdot K_j}) \cdot \mathbb{I}[(x_i, y_i) \in \mathcal{N}_j] \right) \leq B$. According to the fully adaptive composition theorem of individual RDP (Theorem 2.5), by ensuring the sum is less than or equal to $B$ for all time steps $t$ and for all data point $(x_i, y_i)$, the algorithm is shown to be $(\alpha, \alpha \cdot B)$-RDP. □

We show the full algorithm of Ind-KNN-Hash in Algorithm 1.

## 2 MORE EXPERIMENTS

**Ablation study on the size of private data set** We present an ablation study on the size of private dataset, where we simulate private datasets of varying sizes $N \in \{50K, 100K, 200K, 300K, 400K\}$ by replicating the training set of the CIFAR-10 dataset. Figure 1 plots the accuracy of answering $T = 2000$ queries with each private dataset size under $(\epsilon = 1.0, \delta = \frac{1}{N})$-DP with the Resnet50 feature extractor. Both prediction approaches can benefit from an increase in private data: Private kNN can take advantage of subsampling while Ind-KNN could leverage a larger active set. The figure demonstrates that our Ind-KNN scales well with more private data, enabling it to address even millions of queries when the private dataset is billion-scale.

*Accepted for the 39th Conference on Uncertainty in Artificial Intelligence* (UAI 2023).

**Algorithm 1** Ind-KNN-Hash

**Input**: Dataset $S \in (\mathcal{X} \times \mathcal{Y})^n$, number of hash tables $L$ and the width parameter $b$, the kernel function $\kappa(\cdot, \cdot)$, the minimum kernel weight threshold $\tau$, sequence of queries $q_1, ..., q_T$, the noisy scale $\sigma_1$ and $\sigma_2$ and the individual budget $B$.
Initialize individual budget $z_i = B, \forall i \in [n]$.
Construct a LSH family: $\mathcal{F} = (f_1, ..., f_L)$, where $f_\ell : \mathcal{R}^d \to \{0, 1\}^b$.
**for** $t = 1$ to $T$ **do**
  Retrieve the hash set: $\mathcal{F}(q_t)$.
  Update the active set $S = \{(x_i, y_i) | z_i > 0, (x_i, y_i) \in \mathcal{F}(q_t)\}$.
  The selected neighbors: $\mathcal{N}_t := \{(x_i, y_i) | \kappa(x_i, q_t) \geq \tau \text{ for all } i \in S\}$.
  Drop $(x_i, y_i)$ from $\mathcal{N}_t$ if $z_i \leq \frac{1}{2\sigma_1^2}$.
  Release $|\mathcal{N}_t|$: $K_t := |\mathcal{N}_t| + \mathcal{N}(0, \sigma_1^2)$.
  **for** $(x_i, y_i) \in \mathcal{N}_t$ **do**
    Update $z_i$ after releasing $K_t$: $z_i = z_i - \frac{1}{2\sigma_1^2}$.
    Evaluate individual "contribution": $g_i = \min\left(\frac{\kappa(x_i, q_t)^2}{2\sigma_2^2 \cdot K_t}, \sigma_2\sqrt{2K_t z_i}\right)$.
    Update $z_i$ after releasing label: $z_i = z_i - g_i$.
  **end for**
  Compute $a_t = \arg\max_{j \in [c]} \left(\sum_{i \in \mathcal{N}_t} \kappa(x_i, q_t) \cdot y_i + \mathcal{N}(\mathbf{0}, \sigma_2^2 \cdot K_t \mathbf{1}_c)\right)_j$.
**end for**
**Return** $(a_1, ..., a_T)$

**Ablation study on the threshold $\tau$.** The minimum kernel weight threshold $\tau$ determines the number neighbors selected for each query-response pair. We conduct an ablation study to investigate the relation between the optimal $\tau$ and the privacy level. Table 2 provides the set of hyper-parameters of Ind-kNN that results in the best utility. Our finding shows that the optimal choice on $\tau$ increases as $\epsilon$ increases across four datasets and two kernel methods. We conjecture this is because when $\epsilon$ is small, the added noise requires a larger margin among the top-k votes to determine the correct output, thus requiring a smaller $\tau$. In contrast, when $\epsilon$ is large, the smaller noise scale enables the algorithm to pick a set of more selective neighbors, thus resulting in a larger $\tau$.

Table 1: The range of hyper-parameters for Private kNN.

| Hyper-parameters | CIFAR-10 | Fashion MNIST | AG News | DBpedia |
|---|---|---|---|---|
| sampling ratio $p$ | $\{0.02, 0.05, 0.1, 0.2\}$ | | | |
| number of neighbors $K$ | $\{100, 200, 300, 400\}$ | $\{100, ..., 500\}$ | $\{100, ..., 600\}$ | $\{100, ..., 600\}$ |

## 3 EXPERIMENTAL DETAILS

In this section, we present the implementation details of Ind-KNN and Private kNN.

**Hyper-parameters search of Ind-kNN.** The noise scale $\sigma_1$ is set to be $\sqrt{\frac{T}{6B}}$ to use roughly half of the individual RDP budget $B$ for each data point being selected at every query. Further reducing $\sigma_1$ does not result in significant improvement. To prevent overflow in $\frac{\kappa(x_i, q_t)^2}{2\sigma_2^2 K_t}$ due to random noise, $K_t$ is set to $\max(K_t, 30)$ for all experiments. For Ind-KNN using a cosine kernel, we fine-tune the noise scale $\sigma_2$ and the threshold $\tau$ on the validation set. To reduce the computational cost of searching all possible $(\sigma_2, \tau)$-pairs, we first estimate the optimal threshold $\tau$ by running a non-private Ind-KNN on the valid set to collect individual kernels weights and sweep through $\tau \in \{0.05, 0.1, ..., 0.95\}$. With $\tau^*$ in hand, we perform a second round of hyper-parameter search for the optimal $(\sigma_2, \tau)$ pair under different privacy levels, where $\tau$ ranges between $[\tau^* - 0.05, \tau^* + 0.05]$. The table below records the range of $(\sigma_2, \tau)$ pairs consider in the second-round search.

We present the range of hyper-parameters search for Private kNN in Table 1 and the best hyper-parameter sets we use in Table 2.

**Feature preprocessing.** As we mentioned in the experiment section, we use a pre-trained ResNet50 model on ImageNet for feature extraction in image classification tasks. Then we perform L2 normalization on the extracted features as a

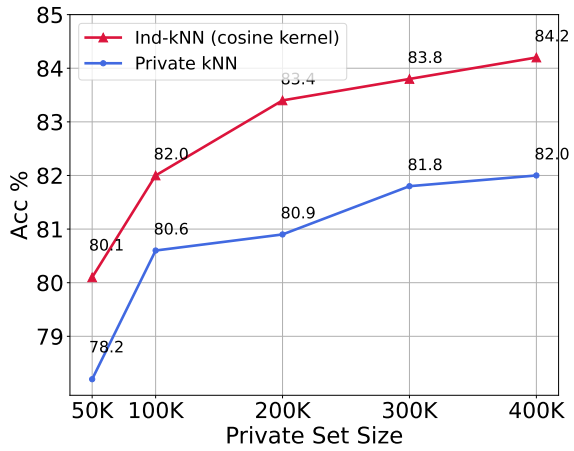


Figure 1: Accuracy of answering $T = 2000$ queries for a private dataset of size $N$ under $(\epsilon = 1., \delta = 1/N)$-DP. We use the ResNet50 pre-trained model as the feature extractor.

preprocessing step. As for text classification, the extracted features from sentence-transformer are already normalized. In this case, we don't need to apply any additional preprocessing steps.

Table 2: Set of hyper-parameters of Ind-KNN resulting the best utility for a set of privacy budgets used in Sec 4.2. We apply the ViT-based feature extractor for CIFAR-10, the pre-trained ResNet50 for Fashion MNIST and the sentence-transformer for AG News and DBpedia.

| Methods | Datasets | $\epsilon = 0.5$ | $\epsilon = 1.0$ | $\epsilon = 1.5$ | $\epsilon = 2.0$ |
|---|---|---|---|---|---|
| Cosine kernel $(\sigma_2, \tau)$ | CIFAR-10 | $(\sigma_2 = 0.7, \tau = 0.26)$ | $(\sigma_2 = 0.4, \tau = 0.26)$ | $(\sigma_2 = 0.4, \tau = 0.27)$ | $(\sigma_2 = 0.4, \tau = 0.28)$ |
| | Fashion MNIST | $(\sigma_2 = 1.3, \tau = 0.6)$ | $(\sigma_2 = 0.6, \tau = 0.6)$ | $(\sigma_2 = 0.3, \tau = 0.6)$ | $(\sigma_2 = 0.3, \tau = 0.6)$ |
| | AG News | $(\sigma_2 = 0.6, \tau = 0.35)$ | $(\sigma_2 = 0.4, \tau = 0.36)$ | $(\sigma_2 = 0.25, \tau = 0.37)$ | $(\sigma_2 = 0.2, \tau = 0.38)$ |
| | DBpedia | $(\sigma_2 = 0.45, \tau = 0.35)$ | $(\sigma_2 = 0.3, \tau = 0.37)$ | $(\sigma_2 = 0.2, \tau = 0.37)$ | $(\sigma_2 = 0.1, \tau = 0.38)$ |
| RBF kernel $(\sigma_2, \tau, \nu)$ | CIFAR-10 | $(\sigma_2 = 0.6, \tau = 0.8)$ | $(\sigma_2 = 0.5, \tau = 0.25)$ | $(\sigma_2 = 0.4, \tau = 0.26)$ | $(\sigma_2 = 0.2, \tau 0.28)$ |
| | Fashion MNIST | $(\sigma_2 = 1.3, \tau = 0.83)$ | $(\sigma_2 = 0.7, \tau = 0.82)$ | $(\sigma_2 = 0.4, \tau = 0.84)$ | $(\sigma_2 = 0.3, \tau = 0.84)$ |
| Hash $(L = 30, b, \sigma_2, \tau)$ | CIFAR-10 | $(b = 8, \sigma_2 = 0.6, \tau = 0.25)$ | $(b = 8, \sigma_2 = 0.4, \tau = 0.50)$ | $(b = 8, \sigma_2 = 0.3, \tau = 0.52)$ | $(b = 8, \sigma_2 = 0.2, \tau = 0.53)$ |
| | AG News | $(b = 9, \sigma_2 = 0.7, \tau = 0.35)$ | $(b = 9, \sigma_2 = 0.4, \tau = 0.36)$ | $(b = 9, \sigma_2 = 0.25, \tau = 0.36)$ | $(b = 9, \sigma_2 = 0.2, \tau = 0.36)$ |

Table 3: The range of hyper-parameters for Ind-KNN.We apply the ViT extractor for CIFAR-10, the pre-trained ResNet50 model for Fashion MNIST and the sentence-transformer for AG News and DBpedia.

| Hyper-parameters | CIFAR-10 | Fashion MNIST | AG News | DBpedia |
|---|---|---|---|---|
| Noise scale $\sigma_1$ | $\sqrt{\frac{T}{6B}}$ | | | |
| Noise scale $\sigma_2$ | $\{0.1, 0.2, ..., 0.9\}$ | | | |
| Minimum threshold $\tau$ (cosine kernel) | [0.25, 0.30] | [0.58, 0.63] | [0.35, 0.40] | [0.35, 0.40] |
| Minimum threshold $\tau$ (RBF kernel) | [0.68, 0.73] | [0.8, 0.85] | - | - |
| scale parameter $\nu$ (with RBF kernel) | $e^{1.5}$ | $e^{1.5}$ | - | - |